



WxDevC++ es una herramienta gratuita que permite desarrollar programas en lenguaje C y C++, ya sea en modo de consola o con interfaz gráfica. Puedes descargar este programa en <http://wxdsgn.sourceforge.net/>.

Este tutorial te servirá para aprender a utilizar el entorno de este software y crear algunos programas de ejemplo. Vamos a comenzar con el lenguaje C.

Una vez abierto el programa, damos clic en File>New>Project. Nos aparecerá una ventana como la siguiente:



Elegimos Console Application y seleccionamos C Project. Escribimos un nombre para proyecto y damos clic en OK. Aparecerá otra ventana para guardar los archivos del proyecto, puedes guardarlos con el mismo nombre. Entonces ya nos aparece el editor de código con un pequeño programa escrito.



Aquí podemos cambiar el texto entre comillas de printf por "Hola Mundo". Para compilar el programa, damos clic en el menú, en Execute>Compile & Run o simplemente presionamos la tecla F9, esto para ejecutar el programa, si sólo requieres compilarlo, pues nada más das clic en Compile.



El programa en ejecución mostrará el mensaje y se quedará así hasta que se presione una tecla o se cierre la ventana. Esta pausa en el programa se realiza con getch().

Como se mencionó al inicio, esta es una guía para el uso de WxDevC++, no un curso de programación, es necesario que sepas ya los conceptos básicos de programación y la sintaxis del lenguaje C, para que no sea necesario estar mencionando qué significa que una variable sea de tipo entero o flotante, que se debe terminar la línea de código con punto y coma,



cómo funcionan los ciclos, entre otros conceptos básicos.

El programa nos muestra un panel llamado Project Inspector, donde se pueden ver los archivos que conforman nuestro programa.



En la pestaña Project, se tiene un elemento con el nombre del proyecto, que alberga los archivos del mismo, como el archivo de código que lleva por extensión la letra c.

También podemos observar otro panel llamado Compiler, que nos muestra advertencias o errores cuando intentamos compilar el programa.



Aquí podemos observar la línea que tiene algún error y una explicación del por qué no funciona.

Si te estás preguntando dónde se almacena el programa ejecutable que se crea al compilar, sólo tienes que buscar la carpeta donde guardaste el proyecto, ahí aparecen todos los archivos del proyecto y una carpeta llamada Output, con otra carpeta dentro llamada MingW, donde encontrarás el exe de tu programa.

Regresando a la programación, además de utilizar las funciones y librerías del lenguaje C, como se utiliza la consola de comandos de Windows para correr el programa, es de esperarse que se puedan utilizar los comandos de la misma consola, esto mediante la función system. Por ejemplo, para la pausa que mencionamos al inicio, se puede sustituir getchar() por system("pause").



Al ejecutar este código, el resultado es el siguiente:



La pausa de la consola de comandos de Windows siempre muestra un mensaje para continuar, y como ya se tenía un texto antes, se muestra todo amontonado. Para evitar esto, agregamos `\n` después de la última letra del texto para que el siguiente texto comience en una línea nueva. Claro que también se puede quitar el texto de la pausa con `>null` o utilizando `getchar()`.



```
#include <stdio.h>
int main(int argc, char *argv[])
{
    system("title Guia DevCpp");
    printf("Hola Mundo\n");
    system("pause");
    return 0;
}
```

Seguramente ya notaste también que la ventana de CMD ya no muestra toda la ruta del programa que se está ejecutando, en lugar de eso, muestra el título del programa, esto mediante la función de cmd `title`.

Si has trabajado con la consola de comandos, ya sabrás todo lo que puedes hacer, aunque si ya dominas muy bien el lenguaje C sólo en ciertas ocasiones requerirás las funciones del sistema.

Otro punto interesante es que WxDevC++ nos permite agregar un icono al ejecutable y la información como versión, descripción, etc. Para esto, nos vamos a `Project>Project Options`,



aparecerá una ventana como la siguiente:



Aquí podemos elegir el icono dando clic en Library, donde aparecerán los iconos que vienen incluidos en el Dev, pero si quieres utilizar tu propio icono, solo das clic en Browse y lo buscas en tu equipo.



Luego, en la última pestaña, llamada Version Info, podemos agregar información al programa. Para esto seleccionamos la opción que dice Include version information in Project. Luego rellenamos los campos en Additional information.

Damos clic en OK y al compilar aparecerá el programa con su icono. Y en la carpeta donde se crea el ejecutable, podrás distinguirlo por el icono y ver la información al acercar el cursor al archivo o al abrir sus propiedades.



## Aplicaciones con interfaz gráfica

Otro tipo de proyecto que WxDevC++ nos permite crear es WxWidgetsFrame, que es una aplicación con widgets como botones, textos, imágenes, barras, entre otros elementos que ayudan a que la aplicación sea más estética.

Esta opción crea los archivos básicos para que el programa funcione y nos da la opción de agregar elementos a la aplicación, la programación es parecida al entorno de Android Studio, pues se pueden elegir los widgets y agregar el código necesario para su funcionamiento.

Entre las opciones de proyecto también aparece Windows Application, pero aquí sólo se trabaja con código, por lo que resulta más complejo, en otro tutorial nos dedicaremos únicamente a esta parte y en otro trabajaremos con todos los widgets, mientras tanto,



veremos los aspectos básicos.



Al crear el nuevo proyecto, nos aparecerá lo siguiente:



Se crearon seis archivos de proyecto, de momento sólo trabajaremos con los dos últimos, Frm.cpp y Frm.wxform.

El cpp nos mostrará el código del programa y los elementos que utilizaremos, y el archivo wxform mostrará el aspecto gráfico del programa.

En este proyecto, aparecerán datos en el panel llamado Property Inspector, estos datos son las propiedades y los eventos de cada elemento o widget, incluyendo la ventana principal del programa.



También aparece la barra de componentes, donde podremos elegir los elementos que vamos a utilizar en el programa.



En caso de no ver esta barra, puedes abrirla en el menú View>ShowPluginPanels>Show Component Palette.

Para crear un Hola mundo aquí es muy simple, sólo arrastramos el elemento StaticText al programa.



En las propiedades de este elemento podemos cambiar el texto, fuente y color de fondo.



Al compilar y ejecutar el programa podemos ver lo siguiente:



Ya que es el único elemento existente en el programa, el color de fondo se aplicó a toda la ventana, pero con poner otro elemento o modificar el color de la ventana se aplica sólo al texto.

En las propiedades del Frame podemos cambiar el tamaño de la ventana del programa, establecer si queremos barra de scroll o si queremos mostrar u ocultar los botones de maximizar y minimizar.

Como ejemplo práctico podemos agregar una barra de progreso WxGauge, y tres botones, cada uno llenará la barra según el valor que se le asigne.



Recuerda que el texto de los botones se modifica en las propiedades de estos, en Property Inspector, donde el texto a modificar está en el campo Label, no confundir con Name, ya que ese es el nombre del botón, no el texto a indicar.

Si damos doble clic sobre el botón, el programa nos dirige al archivo de código, en la sección del botón, donde podemos agregar las acciones que realizará el programa al dar clic en el botón.



El programa crea una variable para cada elemento que se agregue, en caso de que se requieran asignar acciones para dicho elemento. En el caso de la barra de progreso, se creó el código correspondiente.



Se tienen como valores iniciales un rango máximo de 100 y se inicializa en cero.

Vamos a utilizar la línea de código `WxGauge1->SetValue()`; para establecer el valor de cada botón.

```
/*
 * WxButton1Click
 */

void borrar4Frm::WxButton1Click(wxCommandEvent& event)

{
    WxGauge1->SetValue(20);
}

/*
 * WxButton2Click
 */
void borrar4Frm::WxButton2Click(wxCommandEvent& event)
{
    WxGauge1->SetValue(50);
}

/*
 * WxButton3Click
```



```
*/  
void borrar4Frm::WxButton3Click(wxCommandEvent& event)  
{  
    WxGauge1->SetValue(100);  
}
```

Cuando ejecutemos el programa y demos clic en el botón de 50%, el programa hará lo siguiente:



Si queremos agregar un menú al programa, nos dirigimos a la carpeta MenuBars en el panel de componentes y agregamos WxMenuBar al programa.



En las propiedades de este elemento damos clic en Edit Menu Items, y damos clic en los tres puntos que aparecen, se abrirá una ventana donde agregaremos los elementos del menú.



Aquí damos clic en Add Item para agregar el elemento principal, en este caso, Archivo, al dar clic en Apply se guarda y podemos dar clic en Create Submenu, para agregar más elementos dentro del menú Archivo, en este caso Elemento 1 y 2. Se siguen los mismos pasos para más elementos.

Al ejecutar el programa, aparecerá así:







Los elementos del menú no realizarán nada hasta que se modifique el código correspondiente para agregar una acción.

Aquí termina esta guía, en otro artículo estaremos agregando más programas de ejemplo y nos enfocaremos en todos los componentes que se pueden utilizar.

Si tienes algún comentario, duda o quieres que agreguemos algo al siguiente tutorial, déjalo en los comentarios, debajo de esta página.